Please add a grep_v() method to Enumerable that behaves like the opposite of grep().

For example, if Enumerable#grep() was implemented like this:

```ruby
module Enumerable
  def grep pattern
    select { |x| pattern =~ x }
  end
end
```

then Enumerable#grep_v() would be implemented like this (select becomes reject):

```ruby
module Enumerable
  def grep_v pattern
    reject { |x| pattern =~ x }
  end
end
```

The method name "grep_v" comes from the "-v" option passed to grep(1).

Thanks for your consideration.

---

**History**

### #1 - 03/27/2012 03:39 AM - JacksonWillis (Jackson Willis)

```ruby
=begin
I think that this is a good idea.
=end
```

### #2 - 03/27/2012 04:04 AM - trans (Thomas Sawyer)

Can it be an option?

```ruby
grep(pattern, :invert=>true)
```

### #3 - 03/27/2012 05:48 PM - Eregon (Benoit Daloze)

trans (Thomas Sawyer) wrote:

```ruby
Can it be an option?

grep(pattern, :invert=>true)
```

Agreed, adding another method seems overkill (not to mention it does not look right) for such change.

### #4 - 03/28/2012 01:29 AM - sunaku (Suraj Kurapati)

Alright, then let's simplify it further because passing {:invert=>true} seems more like Rails than Ruby:

```ruby
def grep(pattern, invert=false)...
```

```ruby
[*"hello", "world"].grep(/h/, true)  # => [*"world"]
[*"hello", "world"].grep(/h/, false) # => [*"hello"]
[*"hello", "world"].grep(/h/)       # => [*"hello"]
```
What do you think?

#5 - 03/28/2012 01:33 AM - rosenfeld (Rodrigo Rosenfeld Rosas)
"grep pattern, invert: true" is acceptable to me but "grep pattern, true" is not because the code becomes non readable and you'd have to check the API to understand what "true" means. This has nothing to do with Rails vs Ruby.

#6 - 03/28/2012 01:35 AM - rosenfeld (Rodrigo Rosenfeld Rosas)
Of course, after Ruby support named parameters this could be acceptable too and I would always use it by passing the named parameter instead of just "true".

#7 - 03/29/2012 08:14 AM - sunaku (Suraj Kurapati)
I'd rather not wait 2+ years for this issue to be resolved by making this feature dependent on Ruby 2.0's named parameters feature. I'm also not fond of passing an options hash to grep() because, although you say it has nothing to do with Ruby vs. Rails, in my limited experience with Ruby programming, it's uncommon for Ruby core API to accept options hash. In contrast, it's more common to see Rails core API accept options hash.

So I urge you to reconsider my original proposal to add a new grep_v() method. Precedents can be found throughout the FileUtils standard library:

- ln() vs ln_s() vs ln_sf()
- mkdir() vs mkdir_p()
- cp() vs cp_r()
- rm_f() vs rm_r() vs rm_rf()
- chmod() vs chmod_R()
- chown() vs chown_R()

Notice how these are all separate methods; they're not ln(..., :symbolic => true) and mkdir(..., :prefix => true) and so on.

Thanks for your consideration.

#8 - 03/29/2012 12:49 PM - trans (Thomas Sawyer)
If this were for FileUtils then I'd say grep_v is acceptable. But for Enumerable grep_v is not very acceptable. Option parameters are not something only Ruby 2 does. They are very common throughout all Ruby coding, which is why Ruby 2 is going to have better support for them. The fact that Ruby's core does not often use option parameters is partly a fact that Ruby itself lags behind the application of Ruby.

#9 - 03/29/2012 02:23 PM - now (Nikolai Weibull)
On Thu, Mar 29, 2012 at 01:14, sunaku (Suraj Kurapati) sunaku@gmail.com wrote:

Issue #6150 has been updated by sunaku (Suraj Kurapati).

it's uncommon for Ruby core API to accept options hash.

IO.new and IO.open now take an option Hash. I agree with your general sentiment that it's not consistent with the overall Ruby core API, but most of that API was defined well before option Hashes were an established convention.

#10 - 03/29/2012 09:21 PM - Eregon (Benoit Daloze)
Suraj wrote:

it's uncommon for Ruby core API to accept options hash. [...] Precedents can be found throughout the FileUtilities standard library: [...] 

You'll note most of these methods accept an options Hash (e.g.: rm_rf = rm_r list, options.merge(:force => true)).

Also, to add to what Thomas Sawyer said, FileUtils is a CLI(command)-like API, which Enumerable is definitely not (although likely inspired for the "grep" name).

Rodrigo Rosenfeld Rosas wrote:

Of course, after Ruby support named parameters this could be acceptable too and I would always use it by passing the named parameter instead of just "true".

Be aware that keyword arguments as implemented now (if that is what you mean) do not permit this, they just allow an easier syntax to handle arguments and some optimization:

module Enumerable
  def grep(matcher, invert: false)
if invert
  reject { |e| matcher === e }
else
  select { |e| matcher === e }
end
end

%w[1 2 11 22].grep(/1/) # => ["1", "11"]
%w[1 2 11 22].grep(/1/, invert: true) # => ["2", "22"]

#11 - 03/29/2012 09:39 PM - rosenfeld (Rodrigo Rosenfeld Rosas)
Eregon (Benoit Daloze) wrote:

  Rodrigo Rosenfeld Rosas wrote:

  Of course, after Ruby support named parameters this could be acceptable too and I would always use it by passing the named parameter instead of just "true".

  Be aware that keyword arguments as implemented now (if that is what you mean) do not permit this, they just allow an easier syntax to handle arguments and some optimization:

  %w[1 2 11 22].grep(/1/) # => ["1", "11"]
  %w[1 2 11 22].grep(/1/, invert: true) # => ["2", "22"]

  I'm sorry, Benoit, but I didn't get it. It seems to me that they do permit. What did you mean by "they do not permit this"?

#12 - 03/29/2012 09:57 PM - Eregon (Benoit Daloze)
Rodrigo Rosenfeld Rosas wrote:

  I'm sorry, Benoit, but I didn't get it. It seems to me that they do permit. What did you mean by "they do not permit this"?

  Of course, after Ruby support named parameters this could be acceptable too and I would always use it by passing the named parameter instead of just "true".

  With keyword arguments, you have to pass the keyword: grep(matcher, invert: true), grep(matcher, true) is invalid (wrong number of arguments atm).

#13 - 03/29/2012 10:23 PM - rosenfeld (Rodrigo Rosenfeld Rosas)
Oh, I see, I actually prefer this way so that I won't read code from others just passing "true". This is specially a big issue for boolean parameters as they don't mean anything by themselves.

#14 - 03/30/2012 02:11 AM - mame (Yusuke Endoh)
- Status changed from Open to Assigned
- Assignee set to matz (Yukihiro Matsumoto)
- Target version set to 2.0.0

#15 - 03/30/2012 04:23 PM - matz (Yukihiro Matsumoto)
- Status changed from Assigned to Rejected

We are not going to add grep_v. There's possibility of adding :invert option to grep.

Matz.